 # dplyr연습문제

1.1) 변수Wind의 값이 mean(Wind)이상이고 Temp가 mean(Temp)미만이 되는 케이스만을 선택하여 air_sub1에 할당. 변수는 Ozone, Solar.R만 선택.

```
air_sub1 〈- airquality %〉% as.tibble() %〉%
  filter(Wind〉=mean(Wind,na.rm=TRUE),Temp〈mean(Temp,na.rm=TRUE)) %〉%
  select(Ozone,Solar.R)
```

1.2) ari_sub1의 두 변수 Ozone과 Solar.r의 평균값 밑 케이스의 개수를 계산

```
air_sub1 %〉% summarize(Oz_m=mean(Ozone,na.rm=TRUE),SR_m=mean(Solar.R,na.rm=TRUE),n=n())
```

2.1)mtcars의 row name을 변수model로 전환하여 추가 후 3케이스 출력

```
mtcars %〉% as.tibble() %〉% rownames_to_column(var="model") %〉% print(n=3)
```

2.2)변수 model, mpg, cyl, disp, hp, wt, am 만을 선택하여 cars를 만들기

```
cars 〈- mtcars %〉% as.tibble() %〉% rownames_to_column(var="model") %〉%
  select(model,mpg,cyl,disp,hp,wt,am)
```

2.3)변수 disp를 16.4를 곱하여 disp_cc로 만들고, disp 제거

```
mtcars %〉% as.tibble() %〉% mutate(disp_cc=disp*16.4) %〉% select(-disp)
```

2.4)변수 disp_cc를 이용하여 규칙으로 변수 type를 생성후 cars에 추가

```
mtcars %〉% as.tibble() %〉%
  mutate(disp_cc=disp*16.4) %〉%
  select(-disp) %〉%
  mutate(type=if_else(disp_cc〈1000,"Compact",if_else(disp_cc〈1500,"Small",if_else(disp_cc〈2000,"Midsize","Large"))))
```

2.5)변수 am이 1, cyl이 8인 자동차들의 mpg,disp_cc,type의 값을 출력

```
mtcars %〉% as.tibble() %〉%  mutate(disp_cc=disp*16.4) %〉% select(-disp) %〉%
  mutate(type=if_else(disp_cc〈1000,"Compact",if_else(disp_cc〈1500,"Small",if_else(disp_cc〈2000,"Midsize","Large")))) %〉%
  filter(am==1,cyl==8) %〉% select(mpg,disp_cc,type)
```

2.6)변수 cyl의 값에 따라 차 대수 및 mpg,disp_cc,hp,wt의 평균값 출력

```
mtcars %〉% as.tibble() %〉% mutate(disp_cc=disp*16.4) %〉% select(-disp) %〉%
  mutate(type=if_else(disp_cc〈1000,"Compact",if_else(disp_cc〈1500,"Small",if_else(disp_cc〈2000,"Midsize","Large")))) %〉%
  group_by(cyl) %〉%
  summarize(n=n(),mpg_m=mean(mpg,na.rm=TRUE),disp_cc_m=mean(disp_cc,na.rm=TRUE),wt_m=mean(wt,na.rm=TRUE))
```

★★1장 tibble : 개선된 데이터 프레임(부분매칭 불허!)

-chr:문자형, dbl:숫자형, int:숫자형(정수) as.tibble() : Tibble로 생성

-tibble(x=1:3,y=x+1,z=1) :열 단위 생성

-tibble(~x,~y, 1, "a") : 행 단위 생성

-데이터 프레임과의 비교 1)출력 방식의 차이, 2)row names 처리방식,

3)인덱싱 차이 : $변수전체이름

★★2장 dplyr

1.filter(df , 조건) : 조건(〉,〉=,!=,==,&,|,!,%in%)

%in% 사용법 ex) cyl이 6 또는 8 : cyl==6|cyl==8 대신 cyl %in% c(6,8)

ex)mpg의 값이 mpg의 중앙값과 Q3사이에 있는 자동차 선택

```
mtcars %〉% filter(between(mpg,median(mpg),quantile(mpg,probs=0.75)))
```

ex)Ozone또는 Solar.R이 결측값인 관찰값 선택

```
airquality  %〉% as.tibble() %〉% filter(is.na(Ozone) | is.na(Solar.R))
```

2.arrange(df, 정렬기준 1, 기준2,..) 오름차순 디폴트,내림차순:dese(변수)

-is.na(Ozone): 변수 Ozone이 결측값이 케이스가 우선 순위

3.select(df, 변수 또는 문자열 매칭 함수)

-연속된 변수 선택 : 콜론 연산자 : 사용

-문자열매칭 함수 : start_with("x"): x로 시작하는 변수 선택 / ends_with ("x"): x로 끝나는 변수 선택 / contains("x"): x가 포함된 변수 선택 / num_range("x" ,1:10) 안에 ignore.case=FALSE 사용 시 대소문자 구분

변수 배열 변경 ex) iris에서 마지막 변수를 첫 번째 변수로 재배열

-select(iris, Species, everything()) everything()함수 사용

변수 이름 수정할 때 rename()을 쓰고 select와 똑같이 실행(다른변수 제거x)-rename(mtcars_t, Model=rowname)

4.mutate(df, 새로운 변수 생성 표현식)

-새로 생성한 변수만 남기고 다른 변수 제거 할 때 transmute()생성

5.group_by(df, 변수), summarize(df, name=fun)

-summarize에서 사용되는 함수:mean(),sd(),min(),max(),n(),n_distinct()

예제) 1)월별날 수, Ozone에 결측값이 있는 날수 및 실제 측정 된 날 수

```
airquality  %〉% as.tibble() %〉% group_by(Month) %〉%
  summarise(n=n(),obs=sum(!is.na(Ozone)),miss=sum(is.na(Ozone)))
```

2)월별 첫날과 마지막 날 변수Ozone의 값

```
airquality  %〉% as.tibble() %〉% group_by(Month) %〉%
  summarise(first_oz=first(Ozone),last_oz=last(Ozone))
```

 # GGPLOT2 연습문제

1, 패키지 lattice에 있는 데이터 프레임 barley는 미네소타 주 농경학자들이 보리 종류에 따른 수확량의 차이를 비교하기 위해 2년간 경작하여 얻은 자료이다. 설명변수로는 6군데 경작지(site), 10종류 보리(variety), 경작 년도(year)이고 반응변수는 수확량(yield)이다

1.1)세 설명변수의 조합에 따른 수확량의 분포를 알아보는 그래프를 작성

```
lattice::barley %〉% as.tibble() %〉%
  ggplot(mapping=aes(x=yield,y=variety,color=year)) +
  geom_point() + facet_wrap(~site)
```

1.2)보리 종류(variety)에 따른 수확량(yield)의 비교분석에서 경작지(site)와 년도(year)의 효과를 단순반복 처리 한 다음 그래프 작성

```
lattice::barley %〉% as.tibble() %〉%
  ggplot(mapping=aes(x=yield,y=variety,color=site)) +
  geom_point(mapping=aes(shape=year))
```

1.3)각 보리종류(variety)의 평균 수확량을 계산하여 크기순으로 나타내기

```
lattice::barley %〉% as.tibble() %〉% group_by(variety) %〉%
  summarise(variety_m=mean(yield)) %〉% arrange(desc(variety_m))
```

2. mpg의 변수 hwy는 자동차 고속도로 연비를 나타낸다. 범주형 변수인 fl(연료 종류), trans(변속기 종류)에 따른 hwy의 분포를 알아보자

2.1)변수 fl의 종류별 빈도를 구하기

```
mpg %〉% group_by(fl) %〉% summarise(n=n())
```

2.2)변수 fl에서 c,d,e는 제외하고 p와r를 대상으로 상대도수 막대 그래프

```
mpg %〉% filter(fl=="p" | fl=="r") %〉% ggplot(mapping=aes(x=fl,y=..prop..,group=1)) +
  geom_bar()
```

2.3)변수 trans의 종류별 빈도를 구하기

```
mpg %〉% group_by(trans) %〉% summarise(n=n())
```

2.4)변수 trans의 범주를 'auto' 와 'manual' 로 통합한 변수 am생성 후 변수fl과의 관계를 다음과 같은 막대그래프로 나타내보자

```
mpg %〉% filter(fl %in% c("p","r")) %〉% mutate(am=substr(trans,1,nchar(trans)-4)) %〉% ggplot() +
  geom_bar(mapping=aes(x=fl,fill=am),position="fill")
```

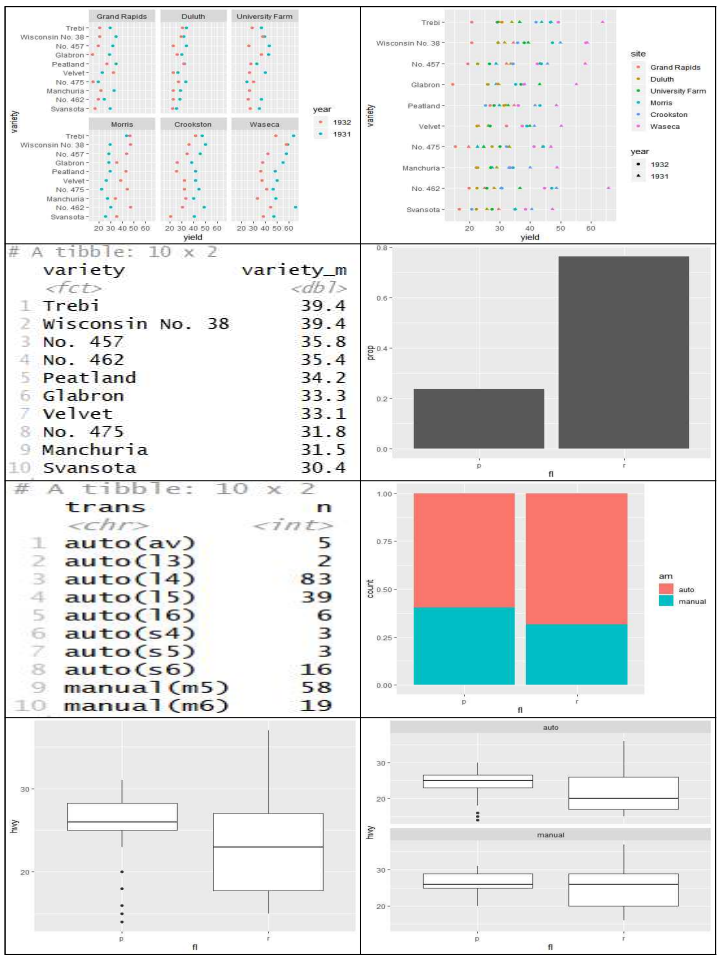
2.5)변수 fl에 따른 hwy의 분포를 상자그림으로 나타내기

```
mpg %〉% filter(fl %in% c("p","r")) %〉%  ggplot() +  geom_boxplot(mapping=aes(x=fl,y=hwy))
```

2.6)변수 am과 f1에 따른 hwy의 분포 비교를 위해 상자그림 작성

```
mpg %〉% filter(fl %in% c("p","r")) %〉% mutate(am=substr(trans,1,nchar(trans)-4)) %〉% ggplot() +
  geom_boxplot(mapping=aes(x=fl,y=hwy)) + facet_wrap(~am,ncol=1)
```

★★3장 GGPLOT2




```
# A tibble: 10 x 2
   variety            variety_m
   <fct>                  <dbl>
 1 Trebi                   39.4
 2 Wisconsin No. 38        39.4
 3 No. 457                 35.8
 4 No. 462                 35.4
 5 Peatland                34.2
 6 Glabron                 33.3
 7 Velvet                  33.1
 8 No. 475                 31.8
 9 Manchuria               31.5
10 Svansota                30.4
```

```
# A tibble: 10 x 2
   trans          n
   <chr>      <int>
 1 auto(av)       5
 2 auto(l3)       2
 3 auto(l4)      83
 4 auto(l5)      39
 5 auto(l6)       6
 6 auto(s4)       3
 7 auto(s5)       3
 8 auto(s6)      16
 9 manual(m5)    58
10 manual(m6)    19
```

기본식 : ggplot(data=<data>) + geom_function(mapping=aes(x=,y=))

mapping는 변수와의 연결, setting는 mapping밖에 전체 조정

ex) 모양 21, 내부색 빨간색, 외곽선색 파란색, 크기 3 외곽선 두께조절 2

```
mpg %>% ggplot() +
  geom_point(mapping=aes(x=displ,y=hwy),shape=21,fill="red",color="blue",size=3,stroke=2)
```

그룹별 그래프 작성 : Facet

: 범주형 변수가 다른 변수에 미치는 영향력을 그래프로 확인하는 방법

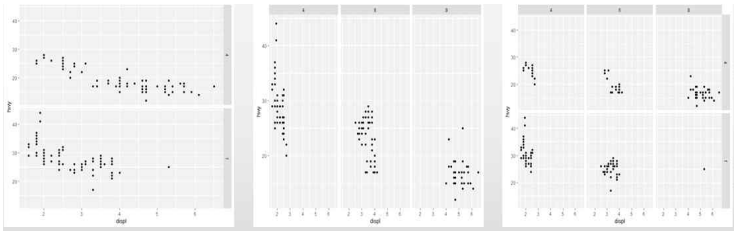
1)facet_wrap(~x): 한 변수에 의한 facet, 열의 수 지정:ncol= ,행의 수 지정: nrow= , 배치순서지정: 행단위 : 디폴트, 열단위 : dir= "v"

2)facet_grid(): 한 변수 또는 두 변수에 의한 facet

-하나의 행으로 배치 : facet_grid(.~x) , 열로 배치 : facet_grid(x~.) 1

-facet_grid(y~x) : 행 범주 : 변수y의 범주, 열 범주 : 변수 x의 범주 2

ex)my_plot + facet_grid(drv~.) , facet_grid(.~cyl), facet_grid(drv~cyl)



-연속형 변수에 의한 faceting : 연속형 변수를 범주형 변수로 변환후 실행

cut_interval(x,n):x를 n개의 같은 길이의 구간으로 구분

cut_width(x,width):x를 길이가 width인 구간으로 구분

cut_number(x,n):n개의 구간으로 구분하되 구간에 속한 개수 비슷하게

ex)데이터 프레임 airquality에서 변수 Ozone,Solar.R,Wind의 관계 탐색

1)변수 Wind를 4개의 구간으로 구분하되 속한 자료의 개수 비슷하게

2)4개의 구간에서 Ozone과 Solar.R의 산점도 작성

```
airquality %>% as.tibble() %>% mutate(group=cut_number(Wind,n=4)) %>%
  ggplot() + geom_point(mapping=aes(x=Solar.R,y=Ozone)) + facet_wrap(~group)
```

-동일 자료에 다른 geom적용(산점도와 비모수 회귀곡선)

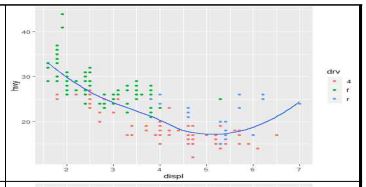
```
mpg %>% ggplot(mapping=aes(x=displ,y=hwy)) + geom_point() + geom_smooth(se=FALSE)
```

-글로벌 매핑 : 함수 ggplot()에서의 매핑, 모든geom함수에 적용

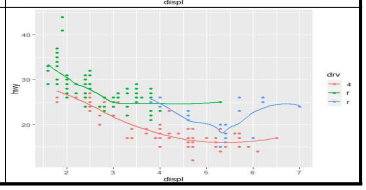
-로컬 매핑 : geom함수에서의 매핑, 해당 geom함수에서만 적용

ex) mpg의 변수 disp과 hwy의 비모수 회귀곡선 작성, 그 위에 산점도 추가하되 drv의 값에 따라 점의 색 구분

```
mpg %>% ggplot(mapping=aes(x=displ,y=hwy)) +
  geom_point(mapping=aes(color=drv)) +
  geom_smooth(se=FALSE)
```

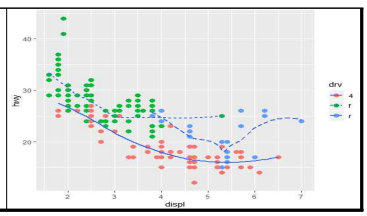


```
mpg %>% ggplot(mapping=aes(x=displ,y=hwy,color=drv)) +
  geom_point() + geom_smooth(se=FALSE)
```



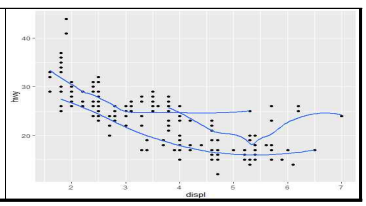
ex)mpg의 변수 displ과 hwy의 비모수 회귀곡선 작성하되 drv에 의해 구분되는 그룹별 추정하여 선의 종류를 다르게 표시. 그 위에 산점도도 추가하되 drv의 값에 따라 점의 색 구분, 점의 크기 확대

```
mpg %>% ggplot(mapping=aes(x=displ,y=hwy)) +
  geom_point(mapping=aes(color=drv),size=3) +
  geom_smooth(mapping=aes(linetype=drv),se=FALSE)
```



ex)변수 drv의 그룹별 따로 비모수 회귀곡선 작성하되, 선의 색과 종류는 같은 것을 사용

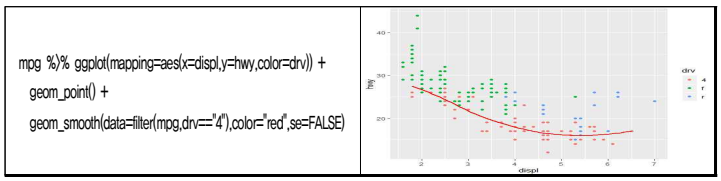
```
mpg %>% ggplot(mapping=aes(x=displ,y=hwy)) +
  geom_point() +
  geom_smooth(mapping=aes(group=drv),se=FALSE)
```



group:그룹을 구성하는 시각적 요소

ex)mpg의 변수 displ과 hwy의 산점도,drv에 따라 점의 색 구분, 비모수 회귀곡선 추가하되 drv가 4인 데이터만을 대상으로 추정

```
mpg %>% ggplot(mapping=aes(x=displ,y=hwy,color=drv)) +
  geom_point() +
  geom_smooth(data=filter(mpg,drv=="4"),color="red",se=FALSE)
```



-통계적 변환 (stat)

산점도 : stat= "identity" , 비모수 회귀곡선 : stat= "smooth" ,

막대 그래프 : stat= "count"

-각 geom함수마다 대응되는 디폴트 stat존재

geom_point() 〉 geom_point(stat= "identity")

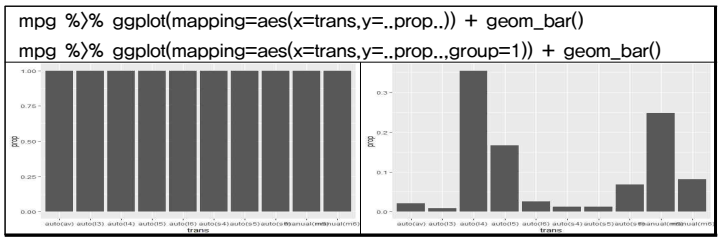
geom_smooth() 〉 geom_smooth(stat= "smooth")

geom_bar() 〉 geom_smooth(stat= "count")

ex)mpg의 변수 trans의 막대 그래프를 상대 도수로 작성

1) 변수..prop..를 이용하여 막대 그래프 작성

```
mpg %>% ggplot(mapping=aes(x=trans,y=..prop..)) + geom_bar()
mpg %>% ggplot(mapping=aes(x=trans,y=..prop..,group=1)) + geom_bar()
```



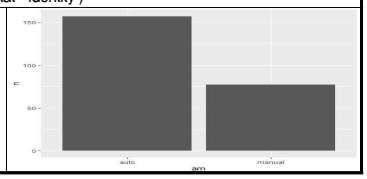
..prop.. : 그룹별 비율, 모든범주를 하나의 그룹구성

2)상대도수 막대 그래프 작성

ex) 도수분포표로 막대 그래프 작성

```
mpg %>% mutate(am=substr(trans,1,nchar(trans)-4)) %>% group_by(am) %>% summarise(n=n()) %>%
  ggplot() + geom_bar(mapping=aes(x=am,y=n),stat="identity")
```

```
# A tibble: 2 x 2
  am        n
  <chr>   <int>
1 auto      157
2 manual     77
```



-위치 조정(점이 겹치는 문제 - jittering)

ex)mpg에서 변수 cty와 hwy의 산점도 작성

-ggplot(data=mpg,mapping=aes(x=cty, y=hwy)) + geom.point()

→ geom_point(position= "jitter")로 문제 해결

-추가되는 난수의 크기 조절 할 경우 geom_jitter()사용

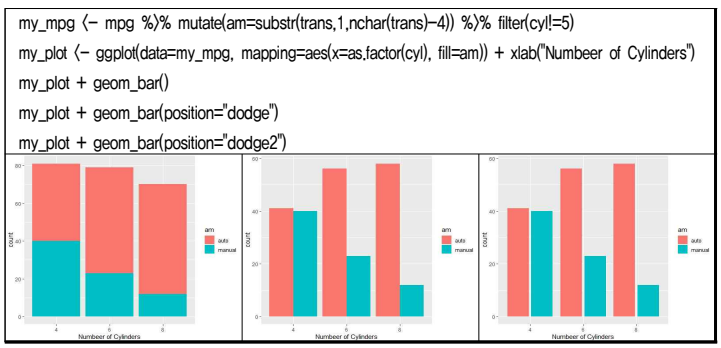
ggplot(data= ,mapping=aes(x= , y=)) +

geom_jitter(width= , height=)

-이변량 막대 그래프 작성

ex)쌓아 올린 막대 그래프와 옆으로 붙여 놓은 막대 그래프 작성

```
my_mpg <- mpg %>% mutate(am=substr(trans,1,nchar(trans)-4)) %>% filter(cyl!=5)
my_plot <- ggplot(data=my_mpg, mapping=aes(x=as.factor(cyl), fill=am)) + xlab("Numbeer of Cylinders")
my_plot + geom_bar()
my_plot + geom_bar(position="dodge")
my_plot + geom_bar(position="dodge2")
```



-나란히 서 있는 상자 그림 geom_boxplot()

ex)그룹을 구성하는 변수가 두 개인 경우의 상자그림

```
mpg %>% mutate(am=substr(trans,1,nchar(trans)-4)) %>%
  filter(cyl!=5) %>%
  ggplot(mapping=aes(x=as.factor(cyl),y=hwy)) +
  geom_boxplot(mapping=aes(fill=am)) +
  xlab("Numbeer of Cylinders")
```